

AD-A082 547

UTAH UNIV SALT LAKE CITY DEPT OF MATHEMATICS  
AN AUTOMATIC QUADRATURE ALGORITHM BASED ON THE  
FEB 80 J SCHWING, K SIKORSKI, F STENGER

F/G 12/1  
TRAPEZOIDAL FORM--ETC(U)  
DAAG29-77-6-0139

ARO-14864.7-M

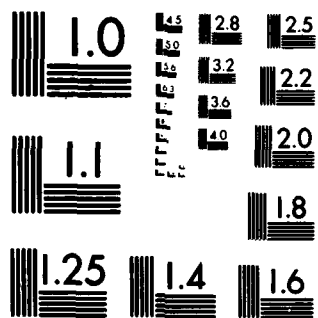
NL

UNCLASSIFIED

1-1  
2-1

1-1

END  
DATE  
FILMED  
5 80  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD A 082547

LEVEL II

18

ARO

19

14864.7-M

R

f

6

AN AUTOMATIC QUADRATURE ALGORITHM BASED  
ON THE TRAPEZOIDAL FORMULA<sup>+</sup>

by

15

DAAG29-77-G-0139

10

J. / Schwing J.\*

K. / Sikorski K.\*\*

F. / Stenger F.\*\*\*

11. 22 Feb 80

12 27

DTIC  
ELECTE

MAR 3 1 1980

A

DISTRIBUTION STATEMENT A

Approved for public release  
Distribution Unlimited

- \* Department of Mathematics, Old Dominion University
- \*\* Department of Mathematics, University of Utah (on leave from the Univ. of Warsaw)
- \*\*\* Department of Mathematics, University of Utah

<sup>+</sup> K. Sikorski and F. Stenger supported by U. S. Army Research Contract Number  
DAAG-29-77-G-0139.

DDC FILE COPY

80 3 26 073

401395

1. INTRODUCTION AND SUMMARY.

This paper describes a FORTRAN subroutine called WNEW, for approximating any one of the following four integrals:

$$(1.1) \quad \int_{-\infty}^{\infty} f(x) dx ;$$

$$(1.2) \quad \int_A^{\infty} f(t) dt \quad (f \text{ not oscillatory, } A \text{ finite}) ;$$

$$(1.3) \quad \int_A^{\infty} F(t) dt \quad (F \text{ oscillatory, } A \text{ finite}) ;$$

$$(1.4) \quad \int_A^B f(t) dt \quad (A, B \text{ finite}) .$$

We remark in view of (1.2) and (1.3) that

$$\int_{-\infty}^B f(t) dt = \int_{-B}^{\infty} f(-t) dt .$$

A description of the parameters of the subroutine WNEW and the method of calling it are given in Sec. 6 of this paper; the user who does not wish to concern himself with the special powers or pitfalls of this subroutine should skip directly to Sec. 6.

The formulas that this subroutine is based on are most powerful when the integrand  $f$  does not have a <sup>^</sup>*singularity* (a singularity is a point where  $df/dx$  does not exist) in the interior of the range of integration; however, singularities at end-points of intervals are allowed. Indeed, it is in the cases where  $f$  has singularities at the end-points of the range of integration

MTS		1
SEC 122		1
Unannounced		1
Justification		1
By _____		
Date _____		
Availability Codes		
1st	Avail and/or special	
A		

that the subroutine WNEW is superior to other subroutines. We concede that Gaussian quadrature which houses the singularities of the integrand in the weight function [1,2] may be superior to the WNEW methods. However, no method [8] is superior to the WNEW method if the exact nature of the singularities at end points of intervals are unknown, or ignored. The subroutines may then be used either directly, or be a part of a polyalgorithm [4,5] which gets accurate results in spite of singularities of the integrand at end-points of an interval of integration.

We emphasize that the WNEW subroutine yields accurate results in spite of singularities at *end points* of an interval. The presence of singularities at interior points of the interval of integrations ~~any~~ considerably slow up the rate of convergence. If the function  $f$  in the integral

$$(1.5) \quad \int_c^d f(t) dt$$

(where  $c$  or  $d$  may be either finite or infinite) has singularities in the interval  $(c,d)$ , then in order to achieve best accuracy we strongly recommend replacing (1.5) by a finite number of integrals with the property that each only has singularities at the end-points of an interval. For example, if the function  $f$  in (1.5) has singularities at  $u$  and  $v$ , where  $c < u < v < d$ , then we recommend replacing (1.5) as follows:

$$(1.6) \quad \int_c^d f(t) dt = \int_c^u f(t) dt + \int_u^v f(t) dt + \int_v^d f(t) dt .$$

Each of these integrals may now be accurately approximated by an appropriate formula used to approximate (1.2), (1.3) or (1.4).

The subroutine WNEW is based on the theory of [6,7]; see also the summary paper [3]. In Sec. 2 we briefly summarize the transformation used to transform each of the integrals (1.2), (1.3) or (1.4) into (1.1), the error when the trapezoidal formula is applied to (1.1), as well as a more accurate description of the type of integrals for which the formulas for approximating (1.1), (1.2), (1.3), and (1.4) are most effective.

Section 3 describes the basis of the algorithm by combining trapezoidal and midordinate rules. In Sec. 4 we give some examples which illustrate the application of the algorithm. In Sec. 5 we illustrate some pitfalls of the algorithm, arising as a consequence of inaccurately computing the integrand near a singularity. We also illustrate methods of circumventing these pitfalls.

In Sec. 6 we give a precise description of the subroutine

WNEW(INTRUL,A,B,EPS,IP)

and of the role of the parameters in this subroutine. We also give a flow-chart description of the main ideas of the subroutine.

In Sec. 7 we give an explicit FORTRAN listing of the subroutine WNEW.

## 2. BASIC IDEAS, TRANSFORMATIONS AND ERROR.

The algorithms of the program are all based on the trapezoidal formula

$$(2.1) \quad \int_{-\infty}^{\infty} f(x) dx \approx h \sum_{k=-\infty}^{\infty} f(kh)$$

where  $h > 0$  is the step size. If  $f$  has no singularities on  $(-\infty, \infty)$  (e.g. if a single formula is used to describe  $f$  on the whole interval  $(-\infty, \infty)$ ) then it may be shown that the error of formula (2.1) satisfies

$$(2.2) \quad |\text{error}| \leq C e^{-c/h}$$

where  $C$  and  $c$  are positive constants that are independent of  $h$ . Thus if  $h$  is replaced by  $h/2$ , then the correct number of significant figures in the approximation (2.1) doubles. Best results are achieved for (2.1) if in addition to being analytic on  $\mathbb{R}$ ,  $f$  also satisfies the inequality

$$(2.3) \quad |f(x)| \leq C' e^{-\alpha|x|}$$

on the real line  $\mathbb{R}$ , where  $C'$  and  $\alpha$  are positive constants. In this case relatively few points are required in the trapezoidal sum to achieve the desired accuracy. If  $f$  decreases to zero at an algebraic rate, as  $t \rightarrow \pm\infty$ , the formula (2.1) is still accurate, however in that case many more points are required to achieve a desired accuracy. This latter situation can sometimes be remedied by use of the transformation

$$(2.4) \quad t = xe^{x^2}$$

and then applying the trapezoidal formula to the transformed integral.

The integral (1.2) is transformed into the integral (1.1) by means of the transformation

$$(2.5) \quad t = A + e^x$$

Applying (2.1) after making the transformation (2.5) results in the quadrature formula

$$(2.6) \quad \int_A^\infty f(t)dt \cong h \sum_{k=-\infty}^{\infty} e^{kh} f(A+e^{kh})$$

(We remark here, that

$$(2.7) \quad \int_{-\infty}^B f(t)dt = \int_{-B}^\infty f(-t)dt \cong h \sum_{k=-\infty}^{\infty} e^{kh} f(B-e^{kh})$$

The formula (2.6) is particularly accurate when used to approximate integrals for which the integrands have an algebraic-type singularity at  $A$ , and which approach zero at an algebraic rate as  $x \rightarrow \infty$ , such as integrals of the form

$$\int_0^\infty t^{-\frac{1}{2}}(t+1)^{-1}dt, \quad \text{or} \quad \int_0^\infty t^{3-\pi}(\log t)(1+t^2)^{-\frac{1}{2}}dt$$

If  $A = 0$ , the ideal boundedness condition on  $f$  corresponding to (2.3) is

$$(2.8) \quad |f(t)| \leq \begin{cases} C't^{\alpha-1} & , \quad 0 < t \leq 1 \\ C't^{-1-\alpha} & , \quad t \geq 1 \end{cases}$$



if (2.3) is satisfied then after making the transformation (2.5) (with  $A = 0$ ) one gets an integral over  $R$  for which the integrand satisfies (2.3).

The integral (1.3) is transformed into the integral (1.1) by means of the transformation

$$(2.9) \quad t = A + \log\{e^x + (1+e^{2x})^{\frac{1}{2}}\}$$

After making the transformation (2.9) and applying (2.1), we get\*

$$(2.10) \quad \int_A^\infty f(t)dt \cong h \sum_{k=-\infty}^{\infty} (1+e^{-2kh})^{-\frac{1}{2}} f(A + \log\{e^{kh} + \sqrt{1+e^{2kh}}\})$$

This formula is best suited for the evaluation of integrals for which the integrand  $f(t)$  has an algebraic-type singularity at  $t = A$  and which behaves in an oscillatory manner as  $t \rightarrow \infty$ . Examples of such integrals are

$$\int_0^\infty t^{-1/3} e^{-t} \cos(3t) dt, \text{ or } \int_0^\infty \log[1 - \frac{\sin t}{t}] e^{-t} dt. \text{ If } A = 0 \text{ in (1.3),}$$

the ideal boundedness condition on  $f$  corresponding to (2.3) is

$$(2.11) \quad |f(t)| \leq \begin{cases} C't^{\alpha-1} & , 0 \leq t \leq 1 \\ C'e^{-\alpha t} & , t \geq 1 \end{cases};$$

\*We recommend care in evaluating  $\log\{e^{kh} + \sqrt{1+e^{2kh}}\}$  when  $e^{kh}$  is small (e.g. if  $e^{kh} \leq .1$ ). In this case, the formula

$$\log\{e^{kh} + \sqrt{1+e^{2kh}}\} = \int_0^{e^{kh}} \frac{dt}{(1+t^2)^{\frac{1}{2}}} = \sum_{j=0}^{\infty} \frac{(\frac{1}{2})_j (-1)^j}{(2j+1)j!} e^{(2j+1)kh}$$

This method of computation is built into the subroutine WNEW.

if  $f$  satisfies (2.11) then after making the transformation (2.9) (with  $A = 0$ ) one gets an integral over  $R$  for which the integrand satisfies (2.3). If  $f$  is oscillatory on  $(0, \infty)$  but does not decrease to zero at the rate (2.11), as in the case of the evaluation of some semi-infinite transforms, such as

$\int_0^\infty t^{-1/3} J_0(at) dt$  the formula (2.10) is still quite accurate. However, in this case a large number of points are required to achieve a desired accuracy. This situation may be remedied\* by an Euler technique, such as that described in [1].

The integral (1.4) is transformed into the integral (1.1) by means of the transformation

$$(2.12) \quad t = \frac{A + Be^x}{1 + e^x}.$$

The boundedness condition on  $f$  corresponding to (2.3) is

$$(2.13) \quad |f(t)| \leq C |(t-A)(B-t)|^{\alpha-1}, \quad A \leq t \leq B,$$

where  $C$  and  $\alpha$  are positive constants. Making the transformation (2.12) and then applying the trapezoidal formula, we get

$$(2.14) \quad \int_A^B f(t) dt \cong (B-A)h \sum_{k=-\infty}^{\infty} \frac{e^{kh}}{(1+e^{kh})^2} f\left(\frac{A+Be^{kh}}{1+e^{kh}}\right).$$

---

\*Another method of circumventing this difficulty is to evaluate

$I(\lambda) = \int_0^\infty e^{-\lambda t} t^{-1/3} I_0(at) dt$  for e.g.  $\lambda = 1/2, 1/4$  and  $1/8$ , and then extrapolate to the limit  $\lambda = 0$ .

### 3. BASIS OF ALGORITHM.

Let us define  $T_h(f)$  and  $M_h(f)$  by

$$(3.1) \quad \begin{cases} T_h(f) = h \sum_{k=-\infty}^{\infty} f(kh) \\ M_h(f) = h \sum_{k=-\infty}^{\infty} f((2k-1)h/2) \end{cases} .$$

Thus the sum on the right hand side of (2.1) is  $T_h(f)$ . Furthermore, it follows that

$$(3.2) \quad T_{h/2}(f) = \frac{1}{2}[T_h(f) + M_h(f)] .$$

Let us start with  $h = 1$  (say) and then compute  $T_h(f)$ . The bound (2.2) shows that  $T_{h/2}(f)$  has at least twice as many significant figures as  $T_h(f)$ . Next, let us compute  $M_h(f)$ , as well as the difference

$$(3.3) \quad |T_h(f) - M_h(f)| = \epsilon/3 \left[ \approx 2 \left[ \int_{-\infty}^{\infty} f(x) dx - T_h(f) \right] \right] .$$

Thus, for sufficiently small  $\epsilon$ ,

$$(3.4) \quad \left| \int_{\mathbb{R}} f(x) dx - T_{h/2}(f) \right| = \left| \int_{\mathbb{R}} f(x) dx - \frac{1}{2}[T_h(f) + M_h(f)] \right| = O(\epsilon^2) < \epsilon .$$

In practice, we cannot sum all of the terms in the infinite sums (3.1). The assumption (2.3) then offers a convenient stopping criteria. Suppose that

we stop the summation in  $T_h(f)$  for  $k > 0$  when

$$(3.5) \quad |f(Nh)| (= O(e^{-\alpha Nh})) < \epsilon/3 .$$

Then we may expect that

$$\begin{aligned} \left| h \sum_{k=N+1}^{\infty} f(kh) \right| &\leq O \left( h \sum_{k=N+1}^{\infty} e^{-kh} \right) \\ (3.6) \quad &= O \left( \frac{h e^{-\alpha(N+1)h}}{1 - e^{-\alpha h}} \right) \\ &= O(e^{-\alpha Nh}) = O(\epsilon) . \end{aligned}$$

That is, we may expect the tail of the series to be of the same order of magnitude as last included term. In order to avoid stopping the algorithm at or near a zero of  $f$  in practice, we make the more reliable test

$$(3.7) \quad |f(Nh)| + |f((N+1)h)| + |f((N+2)h)| < \frac{\epsilon}{3} .$$

Similarly, the sums on the right hand sides of (2.6), (2.10) and (2.14) share the properties of the trapezoidal formula on the right hand side of (2.1), under the assumptions of (2.8), (2.11) and (2.13) respectively, which correspond to the assumption (2.3).

#### 4. EXAMPLES.

The examples of this section illustrate the applications of each of the formulas (2.1), (2.6), (2.10) and (2.14). Each of these formulas may be represented as a single approximating formula

$$(4.1) \quad \int_I f(x) dx \cong h \sum_{k=-K}^L w_k(h) f(z_k(h))$$

The results of various examples are tabulated in Table 4.1. In this table we tabulate the integral to be approximated, the exact value of the integral, the method of quadrature used, EPS, a parameter specified by the user and which is the  $\epsilon$  of the previous section,  $K, L$  (see Eq. (4.1)) and the final approximation achieved.

In the third last and second last entry of Table 4.1 it was not possible to achieve the accuracy of  $\epsilon$ . This phenomenon occurs due to a pitfall in computations; such pitfalls and ways of circumventing these are explained in the next section.

TABLE 4.1 EXAMPLES

INTEGRAL	METHOD USED*	EXACT ANSWER	EPS	K	L	ABSOLUTE VALUE ERROR IN FINAL APPROXIMATION
$\int_{-\infty}^{\infty} e^{-x^2} dx$	Eq. (2.1)	$\sqrt{\pi}$	$10^{-7}$ $10^{-16}$	18 36	18 36	$< 10^{-7}$ $< 10^{-16}$
$2 \int_0^1 \frac{x^{-\frac{1}{2}}}{1+x} dx$	Eq. (2.14)	$\pi =$	$10^{-7}$ $10^{-9}$	78 180	42 108	$< 10^{-7}$ $< 10^{-16}$
$\int_0^{\infty} x^{-\frac{1}{2}} e^{-x} dx$	Eq. (2.10)	$\sqrt{\pi}$	$10^{-16}$	648	312	$< 10^{-16}$
$\int_0^{\infty} \frac{x^{-\frac{1}{2}}}{1+x} dx$	Eq. (2.6)	$\pi$	$10^{-11}$	114	114	$< 10^{-12}$
$\int_0^{\infty} \log[1 - \frac{\sin(x)}{x}] e^{-x} dx$	Eq. (2.10)	-3.045689266 50352	$10^{-14}$	168	132	$< 10^{-16}$
$2 \int_{-1}^1 (3-2x-x^2)^{-\frac{1}{2}} dx$	Eq. (2.14)	$\pi$	$10^{-3}$ $10^{-7}$	12 21	18 36	$< 10^{-4}$ $\approx 10^{-4}$
$2 \int_0^2 \frac{dx}{[x(4-x)]^{\frac{1}{2}}}$	Eq. (2.14)	$\pi$	$10^{-10}$	180	108	$\approx 10^{-9}$
$2 \int_{-\infty}^{\infty} \frac{e^x(1+2e^x)^{\frac{1}{2}}}{(1+e^x)} dx$	Eq. (2.1)	$\pi$	$10^{-16}$	168	324	$< 10^{-16}$

\*All computations were carried out in double precision floating point arithmetic.

### 5. PITFALLS OF COMPUTATION.

The accuracy of the formulas (2.1), (2.6) (2.10) and (2.14) in spite of possible singularities at the end-points of an interval, is based on our being able to accurately evaluate the integrand in a neighborhood of these singularities. This is especially important if (as is often the case for singular integrals) a major contribution to the value of the integral occurs in a neighborhood of the singularity.

The need for exercising care is illustrated by considering the example

$$(5.1) \quad I = 2 \int_{-1}^1 (3-2x-x^2)^{-\frac{1}{2}} dx$$

which is one of the examples in Table 4.1. Direct application of Eq. (2.14) to the approximation of  $I$  results in the formula

$$(5.2) \quad I \approx 4h \sum_{k=-K}^L \frac{e^{kh}}{(e^{kh} + 1)^2} [3-2z_k(h)-z_k(h)^2]^{-\frac{1}{2}}$$

where

$$(5.3) \quad z_k(h) = \frac{e^{kh} - 1}{e^{kh} + 1}$$

The points  $z_k(h)$  cluster near  $x = 1$  (resp. near  $x = -1$ ) for  $k$  large and positive (resp. large and negative). Since  $3-2x-x^2 = 0$  when  $x = 1$ , this results in an error when substituting directly into (5.2) to evaluate this quantity. For example, if  $h = \frac{1}{2}$ ,  $k = 36$  we find, working to 8 significant figures, that  $z_{36}(\frac{1}{2}) = .99999997$ , and that  $[3-2z_k(h)-z_k^2(h)]^{-\frac{1}{2}} = (.00000012)^{-\frac{1}{2}} = 2886.7513$ . On the other hand, if  $x$  is

given by the right hand side of (5.3), we get  $[3-2x-x^2]^{-\frac{1}{2}} = (\frac{1}{2})(e^{kh} + 1)(2e^{kh} + 1)^{-\frac{1}{2}}$   
 $= 2864.8728$ , which is correct to 8 significant figures. We emphasize that the discrepancy is due to the loss of significant figures in the evaluation of  $z_k(h)$  via the use of (5.3).

An additional error occurs in the evaluation of the sum in (5.2). In terms of  $z_k(h)$ , this sum may be written in the form

$$(5.2)' \quad I \cong 2h \sum_{k=-K}^L \{1 - z_k^2(h)\} [(3 + z_k(h))(1 - z_k(h))]^{-\frac{1}{2}}.$$

The trouble occurs in the computer division of two numbers, both of which are close to zero; while the numerator term  $\{1 - z_k^2(h)\}$  expressed in the form  $2e^{kh}/[1 + e^{kh}]^2$  is accurately evaluated, the denominator term is not, since the quantity  $1 - z_k(h)$  only has 1 significant figure of accuracy.

For these reasons it was not possible to achieve an error  $\leq 10^{-5}$  in the evaluation of  $I$  via (5.2).

If we replace  $x$  by  $1 - x$  in (5.1) we get the integral

$$(5.4) \quad J \cong 2 \int_0^2 \frac{dx}{[4x - x^2]^{\frac{1}{2}}}.$$

When the approximated via (2.14), we get the formula

$$(5.5) \quad J \cong 4h \sum_{k=-K}^L \frac{e^{kh}}{(1 + e^{kh})^2} [4x_k(h) - x_k^2(h)]^{-\frac{1}{2}}$$

where

$$(5.6) \quad x_k(h) = \frac{2e^{kh}}{1 + e^{kh}}.$$



The integral  $J$  is the second last integral in Table 4.1. The singularity of the integrand which was at the point  $x = 1$  in (5.1) has now been transformed to the point  $x = 0$  in (5.4). In contrast to the loss of significant figures encountered in the evaluation of  $z_k(h)$  via (5.3), the evaluation of  $z_k(h)$  via (5.1) can be carried out quite accurately. Nevertheless, using double precision, we were still only able to achieve 10 significant figures of accuracy in the approximation of  $J$  via (5.5). The reason for this is the same as that involving the discussion of (5.2)', namely requiring the computer to evaluate the ratio of two very small computed quantities, each having a slight error.

Finally, let us replace  $x$  in 5.1 by  $(e^x - 1)/(e^x + 1)$ . We then get the integral

$$(5.7) \quad H \equiv 2 \int_{-\infty}^{\infty} \frac{e^x}{1 + e^x} (1 + 2e^x)^{-\frac{1}{2}} dx$$

We now approximate  $H$  via (2.1), to get

$$(5.8) \quad H \approx 2h \sum_{k=-K}^L \frac{e^{kh}}{1 + e^{kh}} (1 + 2e^{kh})^{-\frac{1}{2}}$$

In the expression (5.7), the singularity of the integrand has been analytically removed. We thus get the integral in the last entry in Table 4.1. There is now no problem in approximating  $H$  via (5.8) in double precision to get 16 significant figures of accuracy.

## 6. SUBROUTINE FOR AUTOMATIC INTEGRATION.

The FORTRAN subroutine for evaluating one of the integrals (1.1)-(1.4) is called by the statement

WNEW(INTRVL, EPS, A, B, IP) .

### Definitions of Parameters and Required Function Routine.

- (a) General. INTRVL, A, B are used to designate the domain of integration, and which integral of (1.1)-(1.4) is to be evaluated.

EPS is a small positive number, specified by the user. This is the accuracy to which the integral is to be evaluated.

IP is an information parameter.

FN is the name of a user supplied function, having the form FUNCTION FN(X) .

- (b) More detailed descriptions.

INTRVL = 1 means that the integral (1.1) is to be evaluated.  
user sets  $A = B = 0$  .

INTRVL = 2 means that the integral (1.2) is to be evaluated.  
User sets  $A =$  desired numerical value as in (1.2),  $B = 0$  .

INTRVL = 3 means that the integral (1.3) is to be evaluated.  
User sets  $A =$  desired numerical value as in (1.3),  $B = 0$  .

INTRVL = 4 means that the integral (1.4) is to be evaluated.  
User defines the numerical values of  $A$  and  $B$  as in (1.4).

IP is a printout information parameter selected by the user. It is possible to have the following lines printed, depending on values of IP (0, 1 or 2) chosen by the user:

- (i)     H            LOWER        UPPER            T                    M
- (ii)   D9.4           K            L                D20.10            D20.10
- (iii)   CONVERGENCE , THE FINAL APPROXIMATION IS D30.17
- (iv)    DIMENSION EXCEEDED

If IP = 0: all printouts are suppressed;

IP = 1: normal printout occurs. This includes lines (i) (a line of headings), (ii) (one or more lines of numbers under the headings (i)), and one of (iii) or (iv) depending on whether or not convergence is achieved;

IP = 2: only line (iv) is printed, if convergence is not achieved.

Let us briefly explain these parameters in connection with what the program achieves.

Let us denote an arbitrary integral (1.1)-(1.4) by I .

The approximations T and M of I take the form

$$(6.1) \quad T (= T_h) = h \sum_{k=-K}^L w_k(h) f(z_k(h))$$

$$(6.2) \quad M (= M_h) = h \sum_{k=-K}^L w_{2k-1}(h/2) f(z_{2k-1}(h/2))$$

The numerical values of  $H = h$ ,  $K$ ,  $L$ ,  $T$  and  $M$  in line (ii) above are the parameters in (6.1) and (6.2). The integers  $K$  and  $L$  are chosen by the program (e.g. for  $T$ ) such that

$$(6.3) \quad \sum_{k=-K}^{-K+3} w_k(h) |f(z_k(h))| < \text{EPS}/3$$

$$(6.4) \quad \sum_{k=L-3}^L w_k(h) |f(z_k(h))| < \text{EPS}/3$$

Notice that

$$(6.5) \quad T_{h/2} = \frac{1}{2}(T_h + M_h)$$

Convergence occurs, and the printout (iii) follows if the two inequalities

$$(6.6) \quad |T_h - M_h| < \text{EPS}/3$$

and

$$(6.7) \quad K + L + 1 \leq 5000$$

are satisfied. In this case the number  $T_{h/2}$  given by (6.5) is printed out in line (iii). If it is not possible to achieve the requirements (6.3), (6.4) and (6.6) without violating (6.7), the error message (iv) results.

A "summary" flowchart of the integration routine is given on the following page.

## SUBROUTINE WNEW -- FLOWCHART

H=1

EPS3=EPS/3

CALL GEN -- Generate nodes  $x_i(h)$  and corresponding weights  $w_i(h)$  for the upper sum

Find MAXUP so that  $\sum_{i=MAXUP-3}^{MAXUP} w_i(h) |f(x_i(h))| < EPS3$

and compute  $U = \sum_{i=0}^{MAXUP} w_i(h) f(x_i(h))$

CALL GEN -- Generate nodes and corresponding weights for the lower sum

Find MAXLOW so that  $\sum_{i=MAXLOW}^{MAXLOW+3} w_i(h) |f(x_i(h))| < EPS3$

and compute  $L = \sum_{i=MAXLOW}^{-1} w_i(h) f(x_i(h))$

$T = U + L$

CALL GENM - Generate  $x_i(h/2)$  and corresponding weights  $w_i(h/2)$  with  $i$  odd and satisfying  $2 \text{ MAXLOW} + 1 \leq i \leq 2 \text{ MAXUP} - 1$

$M = h \sum_{i=MAXLOW}^{MAXUP-1} w_{2i-1}(h/2) f(x_{2i-1}(h/2))$

$T^* = (T + M)/2$

$|T - M| < EPS3$  Yes

PRINT  $T^*$

STOP

No

$H = H/2$

MAXUP = 2 MAXUP  
MAXLOW = 2 MAXLOW

$T = T^*$

7. FORTRAN LISTING OF WNEW.

In this section we give a detailed FORTRAN listing of the subroutine WNEW. A number of comments are given, which should be helpful to the user of this routine.

SUBROUTINE WNEW(INT, EPS, A, B, IP)

C WNEW IS THE MAIN SUBROUTINE FOR QUADRATURE.

C INT=1 SIGNIFIES AN INFINITE INTERVAL - THE REAL LINE.  
 C THEN SET A=B=0  
 C INT=2 AND INT=3 SIGNIFIES A SEMI-INFINITE INTERVAL  
 C (A.LT.X). THEN SET A AS THE LEFT HAND END POINT AND  
 C B=0.  
 C INT=4 SIGNIFIES A FINITE INTERVAL (A.LT.X.LT.B)  
 C THEN SET A AS THE LEFT AND B AS THE RIGHT END  
 C POINT.  
 C EPS=DESIRED DIFFERENCE BETWEEN THE TRAPEZOIDAL AND  
 C MIDPOINT APPROXIMATIONS.  
 C FN=USER SUPPLIED FUNCTION OF THE FORM 'FUNCTION FN(X)'.  
 C IP=0 SUPPRESSES ALL PRINTOUT.  
 C IP=1 FOR NORMAL PRINTOUT.  
 C IP=2 FOR ERROR INDICATION ONLY.

C WNEW CALLS SUBROUTINE GEN FOR THE INITIAL  
 C CALCULATION OF THE TRAPEZOIDAL APPROXIMATION, AND  
 C THEREAFTER CALLS SUBROUTINE GENM FOR THE APPROXIMATION  
 C OF THE MIDPOINT APPROXIMATION.

IMPLICIT DOUBLE PRECISION (A-H, O-Z)  
 DIMENSION WEIT(5000), VAL(5000)

C INITIALIZE DATA.

IF(IP.EQ.1) TYPE 900  
 H=1.0D0  
 EPS3=EPS/3.0D0  
 SUM1=0.0D0  
 SUM2=0.0D0  
 D=DSQRT(2.0D0)  
 MAXP=48  
 NBEG=1  
 IF(INT.EQ.1) SUM=FN(0.0D0)  
 IF(INT.EQ.2) SUM=FN(A+1.0D0)  
 IF(INT.EQ.3) SUM=FN(A+DLOG(L.0D0+D))/D  
 IF(INT.EQ.4) SUM=(B-A)\*FN((A+B)/2.0D0)/4.0D0

```

C      INITIALIZE THE UPPER TAIL OF THE TRAPEZOIDAL APPROXIMATION
10     CALL GEN(INT,1,NBEG,MAXP,A,B,WEIT,VAL)
      I=0
      CHEK=0.0DO
      M=MAXP
      DO 30 K=NBEG,M
      I=I+1
      EVAL=WEIT(K)*FN(VAL(K))
      D=DABS(EVAL)
      CHEK=CHEK+D
      WEIT(K)=EVAL
      IF(I.LT.3) GO TO 30
      IF(CHEK.GT.EPS3) GO TO 20
      MAXP=K
      NBEG=1
      MAXL=MAX0(MAXP,48)
      GO TO 35
20     CHEK=0.0DO
      I=0
30     CONTINUE
      NBEG=MAXP+1
      MAXP=MAXP+48
      IF((2*MAXP).GT.5000) GO TO 110
      GO TO 10
35     DO 36 K=MAXP,1,-1
36     SUM2=SUM2+WEIT(K)

C      INITIALIZE THE LOWER TAIL OF THE TRAPEZOIDAL
C      APPROXIMATION.
40     CALL GEN(INT,2,NBEG,MAXL,A,B,WEIT,VAL)
      I=0
      CHEK=0.0DO
      M=MAXL
      DO 60 K=NBEG,M
      I=I+1
      EVAL=WEIT(K)*FN(VAL(K))
      WEIT(K)=EVAL
      D=DABS(EVAL)
      CHEK=CHEK+D

      IF(I.LT.3) GO TO 60
      IF(CHEK.GT.EPS3) GO TO 50
      MAXL=K
      GO TO 70
50     CHEK=0.0DO
      I=0
60     CONTINUE
      NBEG=MAXL+1
      MAXL=MAXL+48
      IF((MAXL+MAXP).GT.5000) GO TO 110
      GO TO 40
70     DO 75 K=MAXL,1,-1
75     SUM1=SUM1+WEIT(K)
      APNT=SUM1+SUM2+SUM

```



```

C      COMPUTE THE MIDPOINT APPROXIMATION
      M1=MAXL
      M2=M1+1
      NTRM=MAXP+MAXL
      MAXL=-MAXL
80     NTRN=2*MAXL
      HOV2=H/2.0D0
      CALL GENM(INT,NTRM,NTRN,HOV2,H,A,B,WEIT,VAL)
      SUM=0.0D0
      SUM1=0.0D0
      DO 85 K=1,M1
85      SUM=SUM+WEIT(K)*FN(VAL(K))
      DO 86 K=NTRM,M2,-1
86      SUM1=SUM1+WEIT(K)*FN(VAL(K))
      SUM=SUM+SUM1
      APXN=SUM*H
      TSTR=(APXT+APXN)/2.0D0
      IF(IP.EQ.1)TYPE 901,H,MAXL,MAXP,APXT,APXN
      IF(DABS(APXT-APXN).LT.EPS3) GO TO 100

C      SET UP DATA FOR THE NEXT ITERATION
      H=HOV2
      NTRM=2*NTRM
      MAXL=2*MAXL
      M1=MAXL
      M2=M1+1
      MAXP=2*MAXP
      APXT=TSTR
      IF(NTRM.GT.5000) GO TO 110
      GO TO 80

C      REPORT CONVERGENCE
100    IF(IP.EQ.1)TYPE 902, TSTR
      RETURN

C      REPORT FAILURE TO OBTAIN CONVERGENCE
110    IF(IP.GT.0)TYPE 903
      RETURN
900    FORMAT(7X,'H',7X,'LOWER',3X,'UPPER',11X,'T',19X,'M',/)
901    FORMAT(3X,D9,4,2I8,2D20.10)
902    FORMAT(5X,'CONVERGENCE,THE FINAL APPROXIMATION IS',D30.17)
903    FORMAT(5X,'DIMENSIONS EXCEEDED')
      END

      SUBROUTINE GEN(INT,INF,NBEG,MAX,A,B,WEIT,VAL)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION WEIT(5000),VAL(5000)
      E=2.71828182845904523536D0

C      CALCULATION OF THE WEIGHTS AND NODES FOR THE TRAPEZOIDAL
C      RULE

```

```

C      INFINITE INTERVAL
      IF (INT.GE2) GO TO 10
      IF (INF.EQ.2) GO TO 6
      DO 5 K=NBEG,MAX
      WEIT(K)=1.0D0
      VAL(K)=DFLOAT(K)
5      CONTINUE
      RETURN
6      DO 7 K=NBEG,MAX
      WEIT(K)=1.0D0
      VAL(K)=-DFLOAT(K)
7      CONTINUE
      RETURN

C      SEMI INFINITE INTERVAL
10     WEIT(NBEG)=E**NBEG
      DO 20 K=NBEG+1,MAX
      WEIT(K)=WEIT(K-1)*E
20     CONTINUE
      IF (INF.EQ.1) GO TO 22
      DO 21 K=NBEG,MAX
      WEIT(K)=1.0D0/WEIT(K)
21     IF (INT.EQ.3) GO TO 26
22     IF (INT.EQ.4) GO TO 30
      DO 25 K=NBEG,MAX
      VAL(K)=A-WEIT(K)
25     CONTINUE
      RETURN

26     DO 29 K=NBEG,MAX
      W=WEIT(K)
      POM=DSQRT(W)*DSQRT(1.0D0/W+W)
      IF (W.LT.0.1D0) GO TO 27
      VAL(K)=A+DLOG(W+POM)
      GO TO 28
27     W1=W*W
      W2=(((-429.D0/30720. DO*W1+231. DO/13312. DO)*W1-63. DO/2816.D0)*W1
      W2=((W2+35.D0/1152.D0)*W1-5. DO/112.D0)*W1+3. DO/40.D0)*W1
      VAL(K)=(W2-1.D0/6.D0)*W1+1.D0)*W+A
28     WEIT(K)=W/POM
29     CONTINUE

C      FINITE INTERVAL
30     BMA=B-A
      DO 40 K=NBEG,MAX
      DENM=WEIT(K)+1.0D0
      VAL(K)=(A+B*WEIT(K))/DENM
      WEIT(K)=E*A*WEIT(K)/(DENM+DENM)
40     CONTINUE
      RETURN
      END

```

```

SUBROUTINE GENM(INT,NTRM,NTRN,HOV2,H,A,B,WEIT,VAL)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
DIMENSION WEIT(5000),VAL(5000)

```

```

C      CALCULATION OF THE WEIGHTS AND NODES FOR THE MIDPOINT RULE

C      INFINITE INTERVAL
IF(INT.GE.2) GO TO 20
WEIT(1)=1.0D0
VAL(1)=DFLOAT(L+NTRN)*HOV2
DO 10 K=2,NTRM
WEIT(K)=1.0D0
VAL(K)=VAL(K-1)+H
10  CONTINUE
RETURN

C      SEMI INFINITE INTERVAL
20  EXPH=DEXP(H)
WEIT(1)=DEXP(DFLOAT(1+NTRN)*HOV2)
DO 30 K=2,NTRM
WEIT(K)=WEIT(K-1)*EXPH
30  CONTINUE
IF(INT.EQ.3) GO TO 36
IF(INT.EQ.4) GO TO 40
DO 35 K=1,NTRM
VAL(K)=A+WEIT(K)
35  CONTINUE
RETURN

36  DO 39 K=1,NTRM
W=WEIT(K)
W1=W*W
W3=DSQRT(1.0D0+W1)
IF(W.LT.0.1D0) GO TO 37
VAL(K)=DLOG(W+W3)+A
GO TO 38
37  W2=((-429.D0/30720.D0*W1+231.D0/13312.D0)*W1-63.D0/2816.D0)*W1
W2=((W2+35.D0/1152.D0)*W1-5.D0/112.D0)*W1+3.D0/40.D0)*W1
VAL(K)=((W2-1.D0/6.D0)*W1+1.0D0)*W+A
38  WEIT(K)=W/W3
39  CONTINUE
RETURN

C      FINITE INTERVAL.
40  BMA=B-A
DO 50 K=1,NTRM
DENM=WEIT(K)+1.0D0
VAL(K)=(A+B*(DENM-1.0D0))/DENM
WEIT(K)=BMA*(DENM-1.0D0)/(DENM*DFLOAT(M))
50  CONTINUE
RETURN
END

```

REFERENCES

- [1] Davis, P. and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press (1975).
- [2] Gautschi, W., *Algorithm 331-Gaussian Quadrature Formulas*, CCM ACM II (1968) 432-436.
- [3] Lund, J. R., *The Numerical Evaluation of Transforms*, Ph.D. Thesis, University of Utah (1978).
- [4] Lyness, J. N. and J. J. Kaganove, *A Technique for Comparing Automatic Quadrature Routines*, *Computer Journal* 20 (1976) 170-177.
- [5] Rice, J. R., *A Metalgorithm for Adaptive Quadrature*, *J. ACM* 22 (1975) 61-82.
- [6] Stenger, F., *Remarks on Integration Formulas Based on the Trapezoidal Formula*, *J. Inst. Maths. Applies* 19 (1977) 145-147.
- [7] Stenger, F., *Integration Formulas Based on the Trapezoidal Formula*, *J. Inst. Maths. Applies* 12 (1973) 103-114.
- [8] Stenger, F., *Numerical Methods Based on Whittaker Cardinal, or Sinc Functions*, to appear in SIAM Review.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COVERING BOX
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
AN AUTOMATIC QUADRATURE ALGORITHM BASED ON THE TRAPEZOIDAL FORMULA		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
J. Schwing K. Sikorski & F. Stenger		DAAG-29-77-G-0139
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
University of Utah		
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		February 22, 1980
		13. NUMBER OF PAGES
		25
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
		NA
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
NA		
18. SUPPLEMENTARY NOTES		
The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>This paper describes a FORTRAN program for evaluating one-dimensional integrals of the form <math>\int_{-\infty}^{\infty} f(x)dx</math>, <math>\int_A^{\infty} f(x)dx</math>, <math>\int_A^B f(x)dx</math>, where <math>f</math> may have singularities at end-points of integration.</p>		

